ORUS

Activités plateforme et chaîne image HYP4Uses

PRJ1970-SPE-0034-E2EIS geometry module ATBD A1

	Name	Date and signature
Written by:	**HOULLIER Thomas** Ingénieur d'Etudes et de Recherche - Centre d'Expertise Optique thomas.houllier@sophiaengineering.com	**DD/MM/YYYY**
Verified by:	**Nom Vérificateur** Fonction Vérificateur Email Vérificateur	**DD/MM/YYYY**
Approved by:	**Nom Approbateur** Fonction Approbateur Email Approbateur	**DD/MM/YYYY**
Quality:	**Nom Qualité** Fonction Qualité Email Qualité	**DD/MM/YYYY**

ORUS
RCS Grasse 984 607 671 - Siret 984 607 671 00026
NAF 7490 B - TVA # FR 74 984 607 671

Distribution list

Name	Company
ORUS internal	ORUS

Change log

Name	Version	Date	Subject	Modified Sections/Pages
HOULLIER Thomas	A1	10 Jul 2025	First edition.	-

Table of contents

Table of tables

Table of illustrations

1. OBJECT

This document is the ATBD for the geometry module included in the HYP4Uses E2EIS v3. This module is introduced in the v3 of the simulation, and did not exist in previous versions. The geometry module is tasked with computing the orbit (position and velocity) and attitude of the satellite during an image acquisition.

2. ACRONYMS

Acronym	Meaning
ALT	Along track
API	Application programming interface
ATBD	Algorithm theoretical basis document
AUX	Auxiliary
CCSDS	Consultative committee for space data systems
CSV	Comma-separated file
E2EIS	End-to-end image simulator
ECEF	Earth-centered earth-fixed
ECI	Earth-centered inertial
FFT	Fast Fourier transform
FMC	Forward motion compensation
GCRF	Geocentric Celestial Reference Frame
GNSS	Global navigation satellite system
GSD	Ground sampling distance
IERS	International Earth rotation and reference systems service
ITRF	International Terrestrial Reference System and Frame
LEO	Low Earth orbit
LOF	Local orbital frame
LOS	Line of sight
LVLH	Local vertical local horizontal
MPC	Mission planning center
OBC	On-board computer
PSD	Power spectral density
RAAN	Right ascension at ascending node
ROE	Readout electronics
SOC	Satellite operation center
SSO	Sun-synchronous orbit
SWIR	Shortwave infrared
TDI	Time-delay integration
UTC	Coordinated universal time
VNIR	Visible and near infrared
WGS84	World geodetic system 1984
XML	Extensible markup language

3. APPLICABLE AND REFERENCE DOCUMENTS

3.1 Applicable documents

Title	Reference	Revision	Source	Index
E2EIS architecture and requirements	PRJ1970-SPE-0004	A4	ORUS	AD1

ESA generic E2E simulator Interface Control Document	PE-ID-ESA-GS-464	Issue 1 Revision 4.2 06 Mar 2023	ESA	AD2
Earth Observation Mission Software File Format Specification	PE-ID-ESA-GS-584	Issue 1 Revision 9 13 Dec 2024	ESA	AD3
IERS Conventions (2010)	IERS Technical Note; No. 36 ISSN: 1019-4568	-	IERS Conventions Centre	AD4
Repères instrument	A0003735-SPE-0018	A2	SOPHIA	AD5
Satellite, Operations, SOC and MPC Technical Requirement Specification	PRJ1970-SPE-0001	A3	SOPHIA	AD6
Instrument LoS submodule	PRJ1970-SPE-0047	A1	ORUS	AD7

3.2 Reference documents

Title	Reference	Revision	Source	Index
Transformations between Time Systems	https://gssc.esa.int/navipedia/index.php?title=Transformations_between_Time_Systems&oldid=16123	16123	ESA	RD1
Time References in GNSS	https://gssc.esa.int/navipedia/index.php?title=Time_References_in_GNSS&oldid=16050	16050	ESA	RD2
Earth-centered, Earth-fixed coordinate system	https://en.wikipedia.org/w/index.php?title=Earth-centered,_Earth-fixed_coordinate_system&oldid=1263086982	1263086982	Wikipedia	RD3
International Terrestrial Reference System and Frame	https://en.wikipedia.org/w/index.php?title=International_Terrestrial_Reference_System_and_Frame&oldid=1277563920	1277563920	Wikipedia	RD4
Reference Frames in GNSS	https://gssc.esa.int/navipedia/index.php?title=Reference_Frames_in_GNSS&oldid=15730	15730	ESA	RD5
ITRFVersion	https://www.orekit.org/site-orekit-12.2/apidocs/org/orekit/frames/ITRFVersion.html	12.2	Orekit	RD6
LOFType	https://www.orekit.org/site-orekit-12.2/apidocs/org/orekit/frames/LOFType.html	12.2	Orekit	RD7
NAVIGATION DATA— DEFINITIONS AND CONVENTIONS	CCSDS 500.0-G-4	November 2019	ESA CCSDS	RD8
Geodetic Coordinates	https://en.wikipedia.org/w/index.php?title=Geodetic_coordinates&oldid=1293874486	1293874486	Wikipedia	RD9
Kepler Orbit	https://en.wikipedia.org/w/index.php?title=Kepler_orbit&oldid=1284586488	1284586488	Wikipedia	RD10

ORUS
RCS Grasse 984 607 671 - Siret 984 607 671 00026
NAF 7490 B - TVA # FR 74 984 607 671

Fundamentals of Astrodynamics and Applications	ISBN 978-1881883203	Fourth Edition	David A. Vallado	RD11
ATTITUDE DATA MESSAGES	CCSDS 504.0-B-2	January 2024	ESA CCSDS	RD12
Software ATBD document BIBLOS-3	GMV-BIBLOS3-D5_ATBD	V1.3	GMV / ESA	RD13
Orekit architecture - Time	https://www.orekit.org/site-orekit-12.2/architecture/time.html	12.2	Orekit	RD14

4. MATHEMATICAL NOTATIONS

- Units are referred to in brackets []. We try to give a typical unit even when we only refer to a physical quantity as a whole.
- $[\![a,\ b]\!]$ *with* $a \in \mathbb{Z}, b \in \mathbb{Z}, a \leq b$, is the set of integers from a to b included.
- $\lfloor a \rfloor$, with $a \in \mathbb{R}$, denotes the floor operation.
- $\lceil a \rceil$, with $a \in \mathbb{R}$, denotes the ceil operation.
- $\|a\|$, with $a \in \mathbb{R}^n$, $n \in \mathbb{N}$, denotes the L2 norm of a vector.
- $|a|$, with $a \in \mathbb{R}$, denotes the absolute value.
- $a\%b$, with $a \in \mathbb{R}, b \in \mathbb{R}$, denotes the modulo operator
- $a \cdot b$, with $a \in \mathbb{R}^n$, $b \in \mathbb{R}^n$, $n \in \mathbb{N}$, denotes the dot product between two vectors.

5. INTRODUCTION

The orbital quantities established by the module are distinguished based on the categories:

- *Nominal*, the target quantity, which corresponds to the ideal in the absence of all errors,
- *Real*, the realized quantity. It is the nominal quantity actualized into a realization, taking into account the realization errors.
- *Measured*, the *real* quantity measured by the system's instrument. The measurement is the data which will be known in the real (as opposed to simulated) system. It is the *real* data being measured, taking into account measurement errors.

The present module is concerned only with nominal and real quantities, while the measured quantities will be established in the platform module. Both the *orbit* and *attitude* follow the categories outlined here. These categories are also used in the context of the GMV/ESA simulation software BIBLOS [RD13].

6. ARCHITECTURE

Figure 1 gives the interface of the geometry module. Note the "Orbit/Attitude files" input regroups several files, which are detailed below.

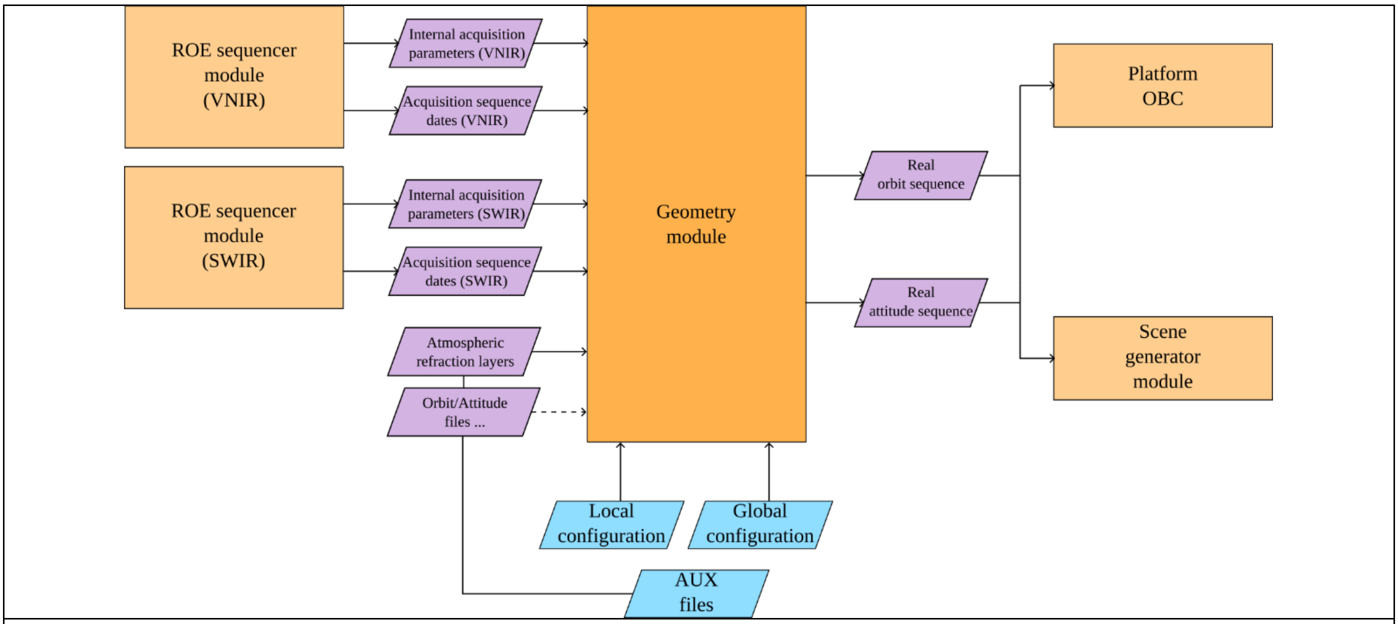

Figure 1 : Geometry module interface

6.1 Input and output data interface

We list exhaustively the input/output files interface of the present module.

6.1.1 Orbit and attitude file formats

The orbit and attitude file formats shall comply with, for input and output, for nominal and real sets, the following formats. These formats are defined by the ESA standard [AD3], dedicated to the ground segment file formats. XML validation schema are available for all of these formats [AD3]. The module shall validate the produced files against the relevant schema systematically. We provide for each format the specific parametrization adopted in the simulation.

6.1.1.1 Orbit state vector file

"Orbit State Vector File" in [AD3] Table 1.

Table 1: Orbit state vector file parametrization

Version	3.0
Ref_Frame	EARTH_FIXED (TBC)
Time_Reference	UTC

6.1.1.2 Attitude quaternion file

"Attitude quaternion file" in [AD3] Table 2.

Table 2: Attitude quaternion file parametrization

Version	3.1
Attitude_File_type	Sat_Nominal_Attitude OR Sat_Attitude
Attitude_Data_Type	Quaternions
Reference_Frame	EARTH_FIXED (TBC)
Time	UTC

6.1.2 Atmospheric refraction layers

The atmospheric refraction is modelled using a multi-layer approach. The atmosphere is decomposed into successive, stacked, layers with their own refractive index. The model is initialized from a csv file with the following characteristics.

- An arbitrary number n_{lay} of layers is defined, corresponding to the number of rows in the file.
- The first column gives the layer index, indexed from 0 to $n_{lay} - 1$.
- The second column gives the lowest altitude at which the layer is applied.
- The third column gives the refraction index of the layer.

The csv file thus follows the format represented in Table 3.

Table 3: Refraction layers file format

Layer index	Lowest altitude (meters)	Refraction index
0	-2000	1.0004
…	…	…
$n_{lay} - 1$	50000	1.000000002

6.1.3 Acquisition sequence dates

The module receives as input either one or two acquisition sequence dates files, corresponding to the VNIR or to the SWIR acquisition sequence. The dates are expressed within the time frame of the satellite using the UTC system. The dates indicate the start of the frames. We note the acquisition dates as,

$$\left\{T_f^{acq} \middle| f \in [\![0, n_f - 1]\!]\right\} \tag{1}$$

The dates are formatted according to the same date format as in the orbit state vector file (section 6.1.1). An example is represented in Table 4.

Table 4: Acquisition sequence dates

Frame index	Frame start date (satellite UTC)
0	2021-06-10T04:57:52.817060
…	…
$n_f - 1$	2021-06-10T04:58:52.817060

6.1.4 Orbit PSD

The module generates a real orbit sequence from the nominal orbit sequence using a PSD of the position error of the satellite. This PSD is a characteristic of the satellite and corresponding position control loop. The PSD is expressed in the satellite frame, along the three cartesian basis axes.

The PSD is given by a table of $n_{psd,o}$ discrete data points, which will be interpolated by the module. The table is contained in a CSV file. The PSD has the following units:

- Frequency [Hz], note the coefficient with zero frequency represents the mean bias, as usual,
- Displacement variance per frequency [m^2/Hz], this is the “power” of the displacement at a particular frequency. In our case this can be seen as a variance of the position error depending on the frequency.

Note that strictly negative frequencies are explicitly disallowed. A non-regular spacing on the frequency axis is allowed.

Table 5 illustrates the contents of such a PSD file.

Table 5: Orbit PSD input file example

Frequency (Hz)	X PSD (m^2/Hz)	Y PSD (m^2/Hz)	Z PSD (m^2/Hz)
0	4.3	2.3	18.2
…	…	…	…
$n_{psd,o} - 1$	0.3	0.8	0.9

6.1.5 Attitude PSD

The module generates a real attitude sequence from the nominal attitude sequence using a PSD of the attitude error of the satellite. This PSD is a characteristic of the satellite and corresponding attitude control loop. The PSD is expressed in the satellite frame, along the three cartesian basis axes (rotation about each of the axes).

The PSD is given by a table of $n_{psd,a}$ discrete data points, which will be interpolated by the module. The table is contained in a CSV file. The PSD has the following units:

- Frequency [Hz], note the coefficient with zero frequency represents the mean bias, as usual,
- Angular variance per frequency [rad^2/Hz], this is the “power” of angular displacement at a particular frequency. In our case this can be seen as a variance of the angular error depending on the frequency.

Note that strictly negative frequencies are explicitly disallowed. A non-regular spacing on the frequency axis is allowed.

Table 6 illustrates the contents of such a PSD file.

Table 6: Attitude PSD input file example

Frequency (Hz)	X PSD (rad^2/Hz)	Y PSD (rad^2/Hz)	Z PSD (rad^2/Hz)
0	4.2e-6	10.3e-6	4.9e-6
…	…	…	…
$n_{psd,a} - 1$	0.9e-6	3.4e-6	1.3e-6

6.2 Algorithmic blocks overview

Figure 2 represents the detailed architecture of the module.

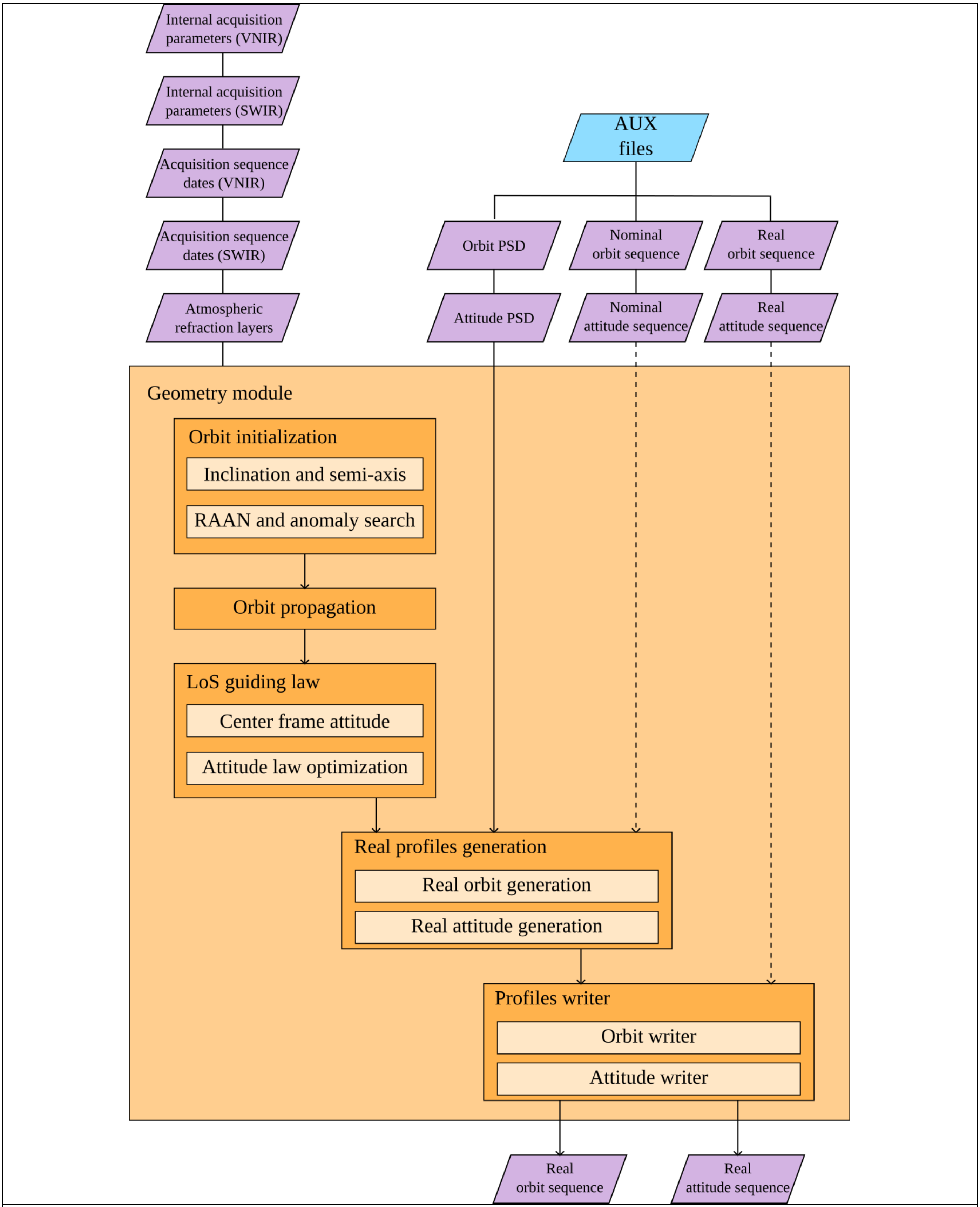

Figure 2 : Geometry module architecture

6.3 AUX files control flow

The module has a control flow related to the following optional input files:

- Nominal orbit,
- Real orbit,
- Nominal attitude,
- Real attitude.

The module's primary objective is to establish the *real* set of orbit and attitude outputs. We also need to be able to manage a user provided set of orbit and attitude files (for instance to test the conformance of the platform supplier attitude law). Note the provided files must be coherent with the acquisition dates and location.
Table 7 indicates explicitly which cases are managed by the module, and which are not. The module shall stop with an error when an untreated case is encountered.

Table 7: AUX files usage

Case index	Nominal orbit	Real orbit	Nominal attitude	Real attitude	Treated?
0	✗	✗	✗	✗	Y
1	✗	✗	✗	✓	N
2	✗	✗	✓	✗	N
3	✗	✗	✓	✓	N
4	✗	✓	✗	✗	N
5	✗	✓	✗	✓	Y
6	✗	✓	✓	✗	Y
7	✗	✓	✓	✓	Y
8	✓	✗	✗	✗	Y
9	✓	✗	✗	✓	Y
10	✓	✗	✓	✗	Y
11	✓	✗	✓	✓	Y
12	✓	✓	✗	✗	Y
13	✓	✓	✗	✓	Y
14	✓	✓	✓	✗	Y
15	✓	✓	✓	✓	Y

We describe how each case is handled:

- Case 0: This is the default case. The nominal orbit and attitude are generated according to the laws described in the present document. The real orbit and attitude are generated from the nominal set by adding a random profile from the PSD files.
- Case 5: No nominal set is computed. The real orbit and attitude are re-interpolated over the acquisition time span and outputted.
- Case 6: The real orbit is reinterpolated over the acquisition time span. The real attitude is computed from the nominal attitude by adding a PSD.
- Case 7: Same treatment as case 5.
- Case 8: The nominal attitude is computed according to the input scene target. The real set is computed by adding a PSD.
- Case 9: The real orbit is computed by adding a PSD on the nominal one. The real attitude is reinterpolated over the acquisition time span.
- Case 10: The real set is computed by adding a PSD.

ORUS
RCS Grasse 984 607 671 - Siret 984 607 671 00026
NAF 7490 B - TVA # FR 74 984 607 671

- Case 11: The real orbit is computed from the nominal one by adding a PSD. The real attitude is reinterpolated over the acquisition time span.
- Case 12: The nominal attitude is computed from the input scene target. The real attitude is computed by adding a PSD. The real orbit is reinterpolated.
- Case 13: The real set is reinterpolated and output.
- Case 14: The real attitude is computed by adding a PSD. The real orbit is reinterpolated.
- Case 15: The real set is reinterpolated and output.

7. CONFIGURATION

We list the parameters used by the module.

- Acquisition parameters (VNIR channel only): Table 8,
- Instrument description (VNIR channel only): Table 9,
- Simulation parameters: Table 10.

Additionally, a strong condition for parameter validity is provided. It is suggested to use them as sanitization of the inputs. Further sanitization is specified in the module, with more specific conditions.

Table 8: Acquisition parameters (VNIR)

Index	Parameter designation	Quantity [unit]	Symbol in this document	Conditions
ACQ1	Spatial binning factor	-	n_{bin}	$[\![1,2]\!]$
ACQ2	Latitude of the scene center to acquire	Latitude [degrees]	D_{lat}	-
ACQ3	Longitude of the scene center to acquire	Longitude [degrees]	D_{lon}	-
ACQ4	Altitude of the scene center to acquire	Altitude [m]	D_{alt}	-
ACQ5	Roll angle of the center LOS at the scene center	Angle [degrees]	D_{roll}	-
ACQ6	Pitch angle of the center LOS at the scene center	Angle [degrees]	D_{pitch}	-
ACQ7	Repeat ground tracks parameters	See below	n, q, m	≥ 0
ACQ8	Orbit eccentricity	Eccentricity [-]	e	≥ 0
ACQ9	Orbit perigee argument	Angle [rad]	pa	-

Table 9: Instrument description (VNIR)

Index	Parameter designation	Quantity [unit]	Symbol in this document	Conditions
IN1	Telescope focal length in the ALT direction	Length [mm]	f_y'	$\neq 0$
IN2	Detector pixel pitch	Length [µm]	p	> 0

Table 10: Simulation parameters

Index	Parameter designation	Quantity	Symbol in this	Conditions

		[unit]	document	
SIM1	Convergence tolerance for the computation of the semi-axis	Length [m]	a_{thresh}	> 0
SIM2	Maximum allowable iterations for the initialization of the orbit	#	$Orbit_{iter}^{init}$	> 0
SIM3	Maximum number of LoS evaluations for the RAAN/anomaly search and attitude optimization	#	f_{ev}	> 0
SIM4	LoS convergence threshold for the RAAN/anomaly search and for the attitude optimization	Distance [m]	D_{tol}	> 0
SIM5	Angular tolerance for the frame alignment during the attitude optimization	Angle [rad]	α_{tol}	> 0
SIM6	Flag enabling the orbit realization errors	Flag	$orbit_{errors}^{p}$	Boolean
SIM7	Flag enable the attitude realization errors	Flag	$attitude_{errors}^{p}$	Boolean
SIM8	Satellite mass	Mass [kg]	sat_{mass}	> 0

8. MODELLING APPROACH

Here are several items specific to the models included in the present module.

- The module must rely heavily on externally validated orbitography software for the definition of usual reference frames, coordinate systems and time systems, as well as for the definition of orbitographic constants. Indeed, the domain-specific parts of orbitography rely on intricate standards which require the use of ephemerides datasets. The implementation and validation of such libraries must remain completely outside the scope of the module. We choose to rely on the Orekit library explicitly. The software versions are as per the requirement E2E-FUN-GM-040 [AD1].
- Note the DEM is never considered in the present module. This is due to a hypothesis of the HYP4Uses mission: the attitude law is established for an ellipsoid only.
- Some parts of the present module may need to be reused in other modules. As such, the implementation may factorize them. The set of ATBD may reflect this factorization in future revisions (TBC).
- Due to the analytical complexity of some parts of the computations, we rely in the present module on numerical search algorithms. The use of robust algorithms (eg. root-finding versus blind minimization) is recommended, and in any case the module must fail if the specified tolerance on the search result is not reached.

An important hypothesis throughout the module is that the primary acquisition sequence is the VNIR one, not the SWIR one. All of the orbit and attitude are of course common to both imaging channels, and we establish everything *as if for a VNIR acquisition*. The SWIR part of the instrument functions with the orbitographic data optimized for the VNIR channel. A consequence of this is that the SWIR acquisition sequence must always be contained in the VNIR acquisition sequence temporally. Note the VNIR channel primacy is enforced even in the cases when the VNIR channel will not produce images, in this case the VNIR sequence is only virtual.

9. DEFINITIONS

We give important definitions and conventions followed by the present module, related to orbitography.

9.1 Orbitographic constants

We list the orbitographic constants which are used throughout the document. The implementer is encouraged to use constants provided in third party libraries rather than copy values manually from other sources, because the exact values can vary depending on the reference systems used. Table 11 lists the constants.

Table 11: Orbitographic constants list

Symbol	Name	Value (Orekit constants)
$\dot{\Omega}_{SunSyn}$	Nodal secular motion caused by J_2 ([rad/s])	$\frac{2\pi}{BESSELIAN\_YEAR}$
$R_\oplus$	Radius of Earth	IERS2010_EARTH_EQUATORIAL_RADIUS
J_2	Zonal harmonic coefficient 2	IERS2010_EARTH_C20
μ	Earth gravitational parameter	IERS2010_EARTH_MU
$\omega_\oplus$	Earth angular velocity	IERS2010_EARTH_ANGULAR_VELOCITY

Note we choose the set of constants values defined by the IERS2010 [AD4].

9.2 Reference frames and Coordinate systems

Figure 3 lists the reference frames in use in the present module, their associated coordinate systems and the data items generated inside them.

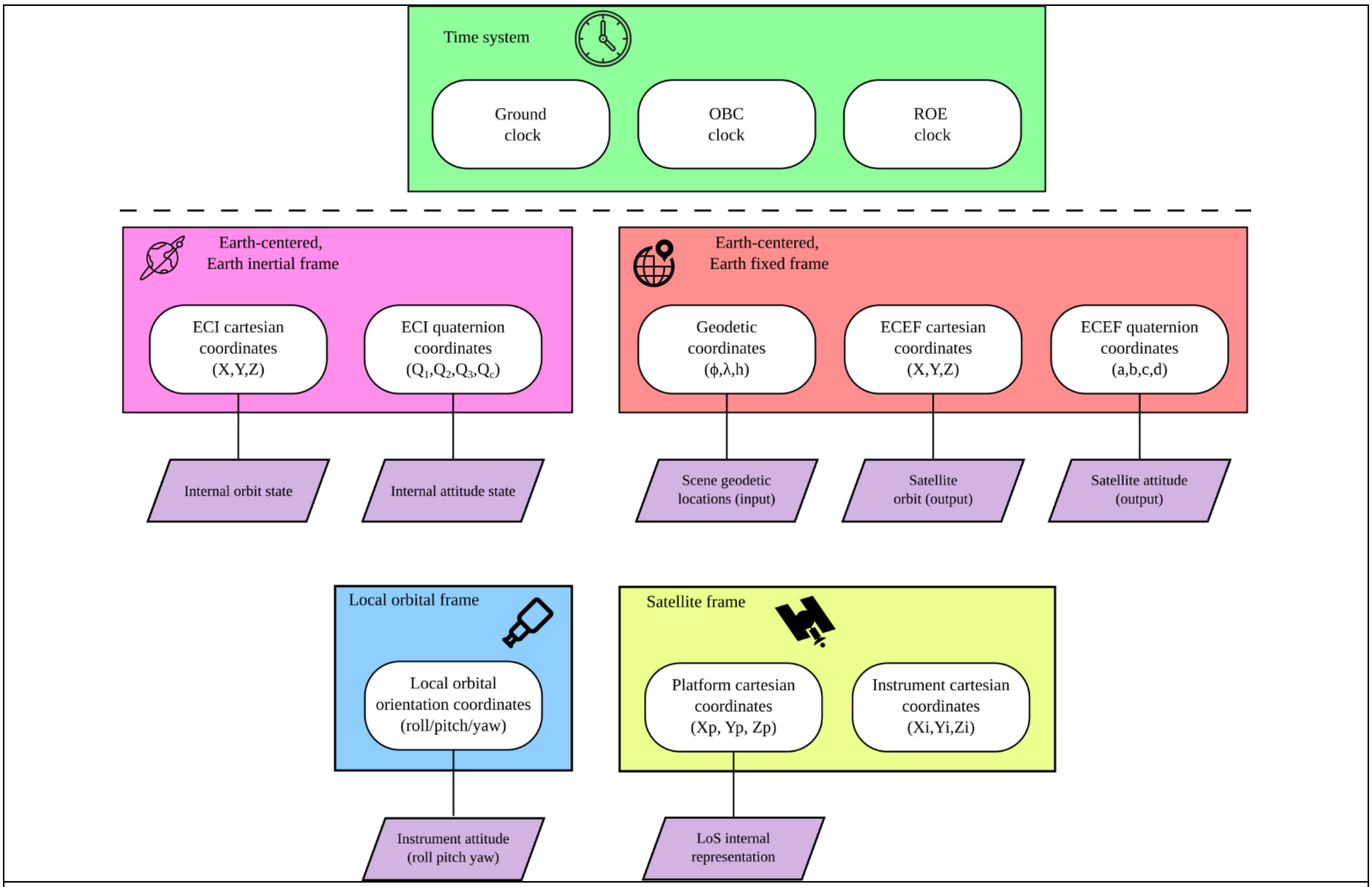

Figure 3 : Frames of reference, coordinate systems and data items

We list the reference frames and coordinate systems of interest. Note that we make a conceptual distinction between reference frame and coordinate system: several coordinate systems may be defined per reference frame. In our present terminology, two coordinate systems are considered to be defined with respect to the same reference frame if the transformation between them is time-independent. For instance, we may use either a polar, geodetic or cartesian coordinate systems with respect to the same ECEF frame, and the transformation between all of them is trivial. However, the transformation between coordinates defined with respect to the satellite frame, and coordinates defined with respect to an ECEF frame is non-trivial and many external parameters must be accounted for in the transformation. This corresponds roughly with the distinction made commonly in general relativity between reference frames.

9.2.1 Frames

We define the reference frames in use. Note that since the precise definition of these frames would be complex, we prefer referring to the Orekit API whenever possible.

9.2.1.1 Time systems

There are many time systems which could be used to date events in the simulation [RD1]. We chose to use **UTC** (Coordinated Universal Time) because of its wide use and compatibility in both spatial and general applications. All major navigational systems, including Galileo are directly synchronized with UTC [RD2].

9.2.1.2 Earth-centered inertial reference frame

We use an Earth-centered inertial reference frame to perform the satellite orbit propagation. The use of such a frame is in fact a hard requirement for many orbitographic propagation tools, including Orekit. We use the **GCRF** (Geocentric Celestial Reference Frame) as our ECI frame.

9.2.1.3 Earth-centered Earth-fixed reference frame

We use an Earth-centered, Earth-fixed reference frame [RD3] (as opposed to celestial/inertial) to describe both the scene locations and satellite orbit and attitude (outputs).
The frame we choose is the latest ITRF [RD4]. This choice is motivated by the fact that the real satellite, in LEO, will get its orbit data from Galileo, which stays aligned with the latest ITRF realization by design [RD5]. The latest ITRF is **ITRF2020**, which we define as the reference frame to use in the simulation. This system is also supported in Orekit [RD6].

9.2.1.4 Satellite frame

The satellite frame is fixed with respect to the satellite body. It allows defining the position of the instrument relative to the satellite. Instrument line of sights are expressed in this frame.

9.2.1.5 Local orbital frame

The local orbital frame rotates with the satellite as it travels along its orbit. The frame is fixed with respect to the local vertical between the Earth center and the satellite center of mass. We use the definition from Orekit **LVLH_CCSDS** [RD7 – RD8]. We reproduce the diagram in Figure 4.

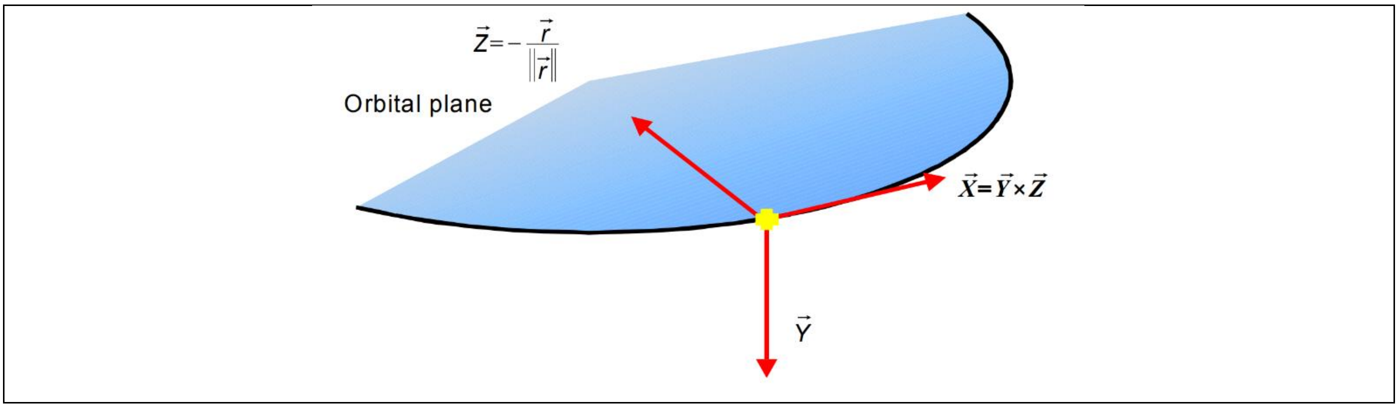

Figure 4 : Local orbital frame diagram [RD8]

9.2.2 Coordinate systems

Coordinate systems are means of representing positions or orientations within a given reference frame.

9.2.2.1 Geodetic coordinate system

We use a geodetic coordinate system [RD9] to refer to scene points. The reference ellipsoid used is WGS84. Figure 5 illustrates this coordinate system.

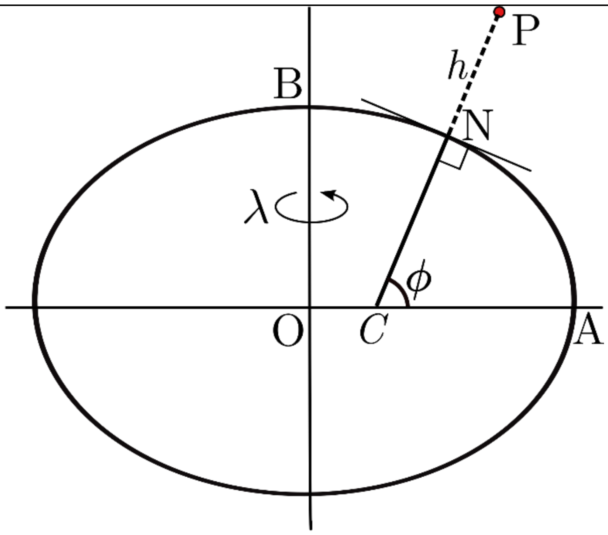

Figure 5 : Geodetic coordinate system [RD9]

9.2.2.2 ECEF coordinate system

We use an ECEF cartesian coordinate system [RD3] to output orbits. This is the natural coordinate system arising from the ITRF reference frame. Figure 6 illustrates this coordinate system.

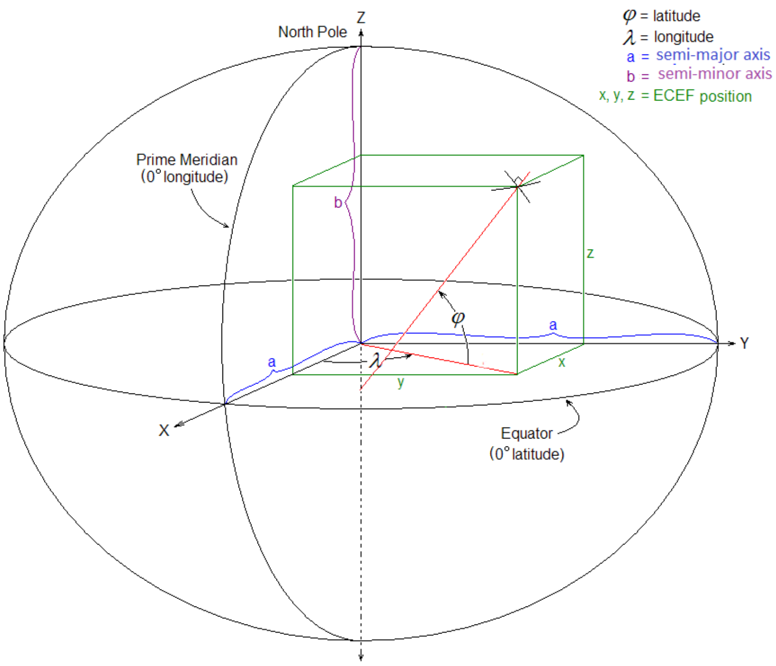

Figure 6 : ECEF coordinate system [RD3]

9.2.2.3 ECEF quaternions

We use the quaternions embedded in the ECEF cartesian coordinate system to treat attitudes.

9.2.2.4 Platform coordinate system

The platform coordinate system is cartesian, defined in the satellite frame with,

- Origin: the center of mass of the satellite.
- X axis: Aligned with the platform nominal ALT motion.
- Z axis: Aligned with the nominal nadir pointing direction.
- Y axis: Completes the right-handed orthonormal coordinate system with X and Y.

This coordinate system is TBC, pending a definition by the platform supplier. It is important that the origin of this platform coordinate system be the center of mass since we need to transform line of sight directions from the platform frame to the local orbital frame. The LoS positions and directions computed in the LoS instrument submodule [AD7] are expressed in this coordinate system.

9.2.2.5 Local orbital roll and pitch

The roll and pitch angles are defined in the local orbital frame, with respect to the usual cartesian coordinate system based on LVLH (CCSDS). Figure 7 illustrates the coordinate system. We use these roll and pitch angles for the initialization of the orbit, but prefer expressing full attitude in quaternions. Note that yaw is not considered here. Note that our definition of roll and pitch angles is independent of the order of rotations, as we do not define rotations but rather LoS angles. Also, since we define the angles with respect to the LOF, considering the scene altitude is not needed.

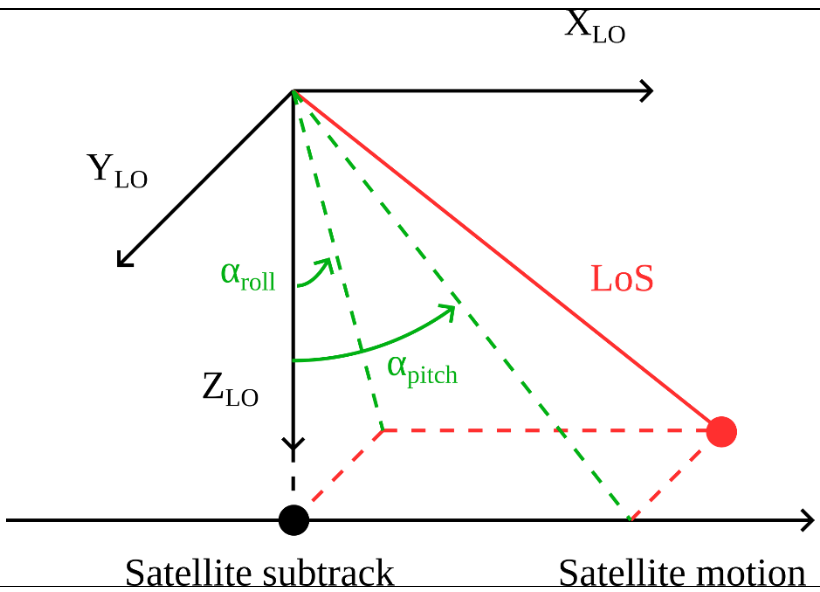

Figure 7 : Illustration of local orbital frame roll and pitch angles

The roll and pitch angles of a given LoS vector in local orbital frame are defined as,

$$\alpha_{roll} = -\operatorname{atan}\left(\frac{\overrightarrow{LoS} \cdot \overrightarrow{Y_{LO}}}{\|\overrightarrow{LoS}\|}\right) \quad (2)$$

$$\alpha_{pitch} = \operatorname{atan}\left(\frac{\overrightarrow{LoS} \cdot \overrightarrow{X_{LO}}}{\left\|\overrightarrow{LoS}\right\|}\right)$$

10. ALGORITHMIC BLOCKS DESCRIPTION

This chapter provides the technical description of processing to the lowest available degree of abstraction.

10.1 Input files validation

All of the orbit and attitude files given as optional inputs must be validated against the XML schema available in [AD3].

10.2 Generic module components

The following components are re-used in multiple parts of the module.

10.2.1 Orbit propagator

TODO: introduce a fine or rough setting (for acquisition or long plot).

Given orbital parameters (which include a position at a given date), the orbit propagator computes the position and velocity of the satellite at nearby dates using orbital mechanics.

The orbit propagator must use the following model:

- The computations must happen within the inertial frame of reference (section 9.2.1.2).
- We use a numerical propagator (as opposed to a simple analytical keplerian propagator) to account for the J2 effect of SSO orbits. This is technically not needed for propagating orbits during the length of a given acquisition, but it is needed to plot representative repeat tracks over long periods.
  - The integrator is Dormand-Prince-853.
    - The minimum step is set to the frame time divided by 10.
    - The maximum step is set to the frame time.
    - The position tolerance is set to the instrument GSD divided by 100.
  - The satellite mass sat_{mass} is as configured in the module parameters.
  - The gravity is modelled with normalized spherical harmonics of degree 8 and order 8.
  - The force model is the Holmes-Featherstone attraction model.

10.2.2 LoS intersector

The LoS intersector computes the intersection of a given LoS (position and direction in the local orbital frame at a given date) with the ellipsoid at a given altitude. It accounts for detailed effects beyond the ideal geometry of the problem. The LoS intersector must use the following model,

- Terrain model: the ellipsoid used in our system of reference (section 9.2.2.1),
- The speed of light is compensated,
- The “aberration of light” velocity is compensated,
- The atmospheric refraction is taken into account using a multi-layer model initialized from the corresponding configuration file. The chromatism is ignored.

10.2.3 Detector pixels of interest

For the purposes of establishing the orbit and attitude, we require the use of LoS intersections of detector pixels with the ellipsoid. We use two different detector positions, referring to the interface defined in the LoS submodule [AD7].

1. The detector center (center of the field of view), at coordinates $(0,0)$ on the detector, named P_{cen}.
2. The detector “above” position, which corresponds to the detector position one pixel above the center, which observed the scene first, at coordinates $(0, p_{above})$, named P_{above}

$$p_{above} = sign(f_y') \cdot p \cdot n_{bin} \qquad (3)$$

With,

- f_y' the instrument ALT focal length in the considered channel,
- p the detector pixel pitch,
- n_{bin} the acquisition binning factor.

10.3 Orbit initialization

We aim at determining the Kepler elements of the satellite for the position at which the center of the scene is observed. The orbit will be propagated backward and forward from this point.

10.3.1 Model

The orbit is described using the common Keplerian elements,

- a: semi-major axis (quantity is length)
- e: eccentricity
- i: inclination (quantity is angle)
- pa: perigee argument (quantity is angle)
- $raan$: right ascension of ascending node (quantity is angle)
- v: anomaly (quantity is angle).

Importantly, note a sun-synchronous orbit is not Keplerian, we only describe one position using the Keplerian elements, but it does not follow that the orbit propagation will necessarily be Keplerian. Some of these Keplerian elements are given directly as module input parameters, the rest are computed as outlined in the present section. Figure 8 illustrates the Keplerian elements.

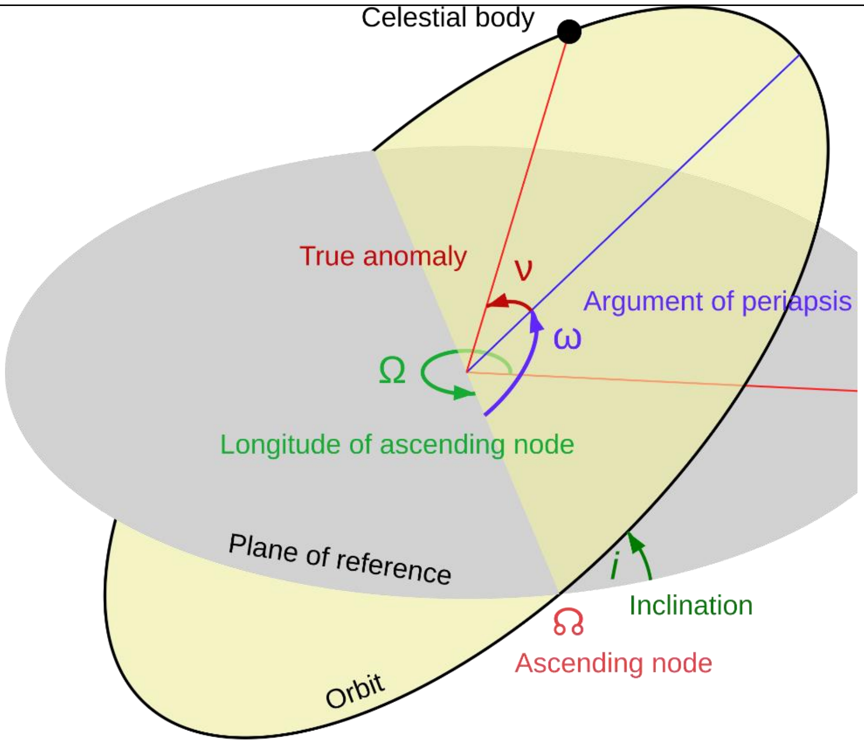

Figure 8 : Illustration of Keplerian elements [RD10]

10.3.2 Input parameters

The orbit (position and velocity of the satellite with respect to time) for an acquisition is first initialized according to the following parameters. The acquisition parameters specify,

- $D = (D_{lat}, D_{lon}, D_{alt})$ the center of the scene to acquire in geodetic coordinates [RD9]: latitude, longitude and altitude above the ellipsoid.
- D_t, the date at which the center of the scene is observed by the nominal instrument LoS. This is expressed as a satellite UTC time. This date is determined from the *acquisition sequence dates* file.

The simulation parameters specify,

- D_{roll} the instrument attitude roll angle at which the center of the scene to acquire is observed,
- D_{pitch} the instrument attitude pitch angle at which the center of the scene to acquire is observed.

The orbit is assumed to be:

- Sun-synchronous,
- Ascending (images are taken in the ascending part of the orbit).

The orbit description parameters specify,

- n: the number of complete orbits per day (integer),
- q: the cycle length (in days),
- m: the additional number of complete orbits after q days,
- e: eccentricity,
- pa: perigee argument.

10.3.3 Acquisition sequence center frame

The center frame in the acquisition sequence is of special interest to the orbitographic computations in the present module. The center frame index is determined from the VNIR acquisition sequence dates (section 6.1.3). There are n_f frames in the sequence. The center frame index in the sequence is noted f_{cen}, and it is determined through,

$$f_{cen} = \left\lfloor \frac{n_f}{2} \right\rfloor \tag{4}$$

The date D_t at which the center frame is acquired is thus read from the VNIR *acquisition sequence dates* file at the index f_{cen}.

10.3.4 Parameters sanitization

We perform a quick sanitization of the parameters. The following conditions must be met, otherwise the module emits an *error*.

$$-90\ deg \leq D_{lat} \leq 90\ deg \tag{5}$$

$$-180\ deg \leq D_{lon} \leq 180\ deg \tag{6}$$

$$-10\ km \leq D_{alt} \leq 10\ km \tag{7}$$

The scene location is constrained to make sense as a point on the Earth. Constraining the altitude allows a more robust determination of the LoS guiding law.

$$year\ 2000 \leq D_t \tag{8}$$

The condition on simulation date is set so that we are within a safe date range with respect to existing time systems [RD14].

$$-85\ deg \leq\ D_{roll} \leq 85\ deg \tag{9}$$

$$-85\ deg \leq D_{pitch} \leq 85\ deg \tag{10}$$

We avoid pathological angles with the conditions on roll and pitch. Note the conditions may not be sufficient for the LoS to always hit the Earth depending on the satellite altitude.

$$n \in \mathbb{N} \tag{11}$$

$$m < q \tag{12}$$

We enforce the definition for the n, q, m repeat ground track parameters.

10.3.5 SSO inclination

The *inclination* orbital parameter i is constrained in the case of SSO orbits, it is given by [RD11 eq.(11-15)], given parameters a and e,

$$i = \arccos\left(-\frac{2a^{7/2} \cdot \dot{\Omega}_{SunSyn} \cdot (1-e^2)^2}{3R_{\oplus}^2 \cdot J_2 \cdot \sqrt{\mu}}\right) \tag{13}$$

Note we do not yet know a, this computation is applied in the next section as part of an iterative algorithm.

10.3.6 Semi-axis for repeatable ground track orbit

Given the initial eccentricity e, SSO inclination constraint and repeat parameters n, q, m, we can deduce the required semi-axis a for an orbit with repeatable ground track. [RD11 algorithm 71 first approach] gives a method to determine the orbital parameters, which we reproduce here.
First, the repeat parameters are computed. The number of *revolutions to repeat* is,

$$k_{rev2rep} = n \cdot q + m \tag{14}$$

The number of *days to repeat* is,

$$k_{day2rep} = q \tag{15}$$

The method is applied with no explanation. It consists in an initialization step and an iterative step. The initialization step computes the following values,

$$k_{revpday} = \frac{k_{rev2rep}}{k_{day2rep}} \tag{16}$$

$$n' = k_{revpday} \cdot \omega_{\oplus} \tag{17}$$

$$a_{new} = \sqrt[3]{\frac{\mu}{n'^2}} \tag{18}$$

$$\Delta\lambda_{rev} = \frac{2\pi \cdot k_{day2rep}}{k_{rev2rep}} \tag{19}$$

Next, the iterative part of the method is repeated until the value of a changes by less than a_{thresh} (as set in the configuration) or the maximum number of iterations $Orbit_{iter}^{init}$ is reached.

$$\begin{gathered}
iter = 0 \\
DO \\
iter \leftarrow iter + 1 \\
a \leftarrow a_{new} \\
p \leftarrow a \cdot (1 - e^2) \\
i \leftarrow Eq.(13) \\
\dot{\Omega} \leftarrow -\frac{3n' \cdot J_2}{2} \cdot \left(\frac{R_\oplus}{p}\right)^2 \cdot \cos(i) \\
\Delta\lambda_{period} \leftarrow \frac{2\pi}{n'} \cdot \dot{\Omega} \\
\Delta lon \leftarrow \Delta\lambda_{rev} + \Delta\lambda_{period} \\
n' \leftarrow \frac{2\pi \cdot \omega_\oplus}{\Delta lon} \\
a_{new} \leftarrow \sqrt[3]{\frac{\mu}{n'^2}} \\
WHILE\ |a_{new} - a| > a_{tresh}\ AND\ iter < Orbit_{iter}^{init} \\
a = a_{new}
\end{gathered} \tag{20}$$

If the maximum number of iterations is reached, the module emits an *error*.

10.3.7 Right ascension of ascending node and anomaly

There are two remaining degrees of freedom to determine the orbit: the RAAN and anomaly. We determine them by a search with the following constraints and objectives:

- The scene center point geodetic coordinates and date are given by the user,
- The roll and pitch angles of the satellite in its local orbital frame are given by the user,
- The LoS intersection with the scene is given by section 10.2.2.

Given these, and the previously determined orbital parameters, the RAAN and anomaly which make the LoS intersect the scene at the center when the satellite roll and pitch are as given are unique.

We use the ground distance between the current LoS intersection and the target LoS as the merit function for our search. The RAAN and anomaly are the search parameters. In other words, we are looking for,

$$(\mathrm{raan}, \nu) = \underset{raan,\nu}{\mathrm{argmin}}\left(\overrightarrow{D_{LoS}}\right) \tag{21}$$

With $\overrightarrow{D_{LoS}}$ the distance between the scene center point (geodetic) and the actual center LoS intersection with the ellipsoid. Figure 9 illustrates the orbit initialization search.

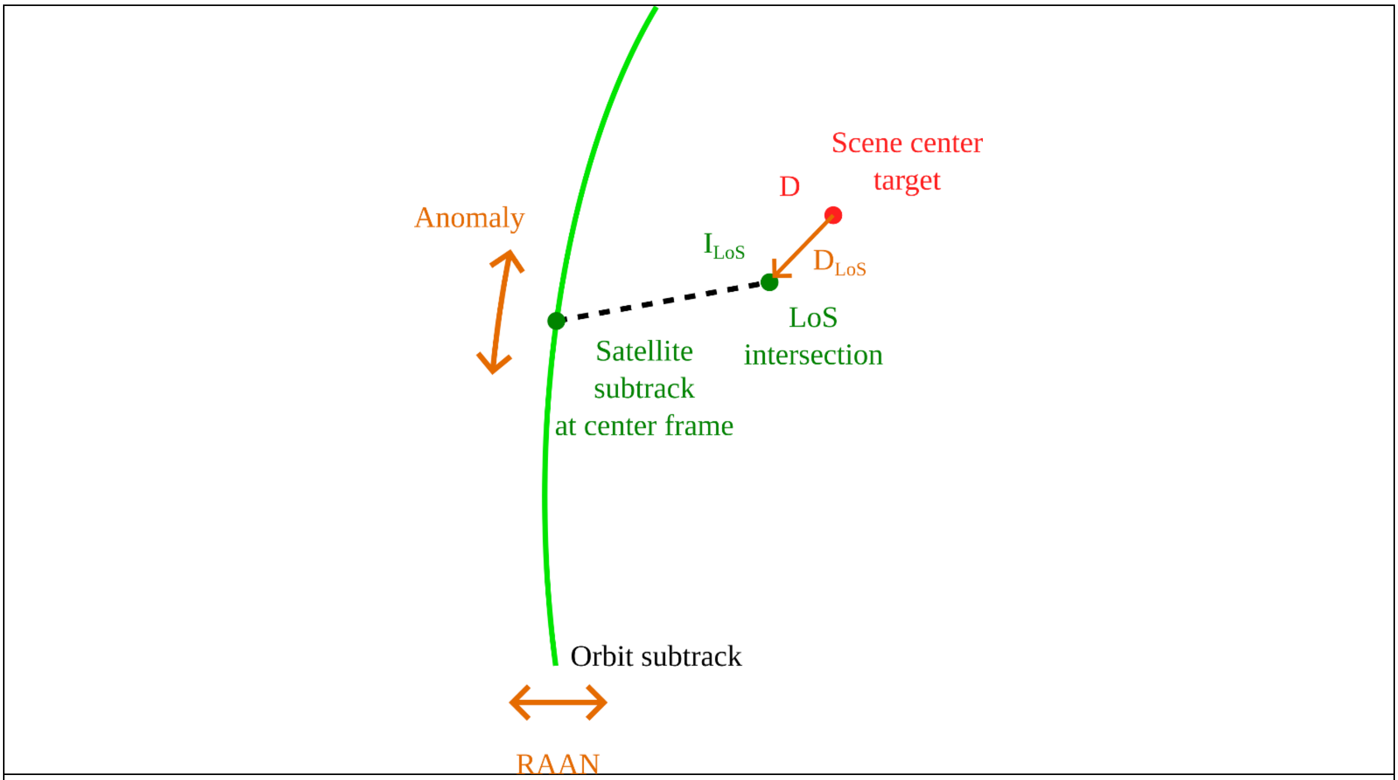

Figure 9 : Illustration of the RAAN and anomaly search for initial orbit

10.3.7.1 Ground distance to target

The distance $\overrightarrow{D_{LoS}}$ between the current LoS intersection I_{LoS} and the scene center target D, is computed following appendix 13.1),

$$\overrightarrow{D_{LoS}} = \overrightarrow{DI_{LoS}} \tag{22}$$

10.3.7.2 RAAN and anomaly initial guess

We need a starting point for our search method, $(raan_{guess}, v_{guess})$ in order to make the search robust. This starting point is computed as follows.

<u>RAAN guess</u>

We obtain the RAAN starting point by computing the angle between the scene position and the X direction of the cartesian coordinate system of the inertial frame. This is exact if the scene is positioned in the XY plane of the coordinate system, and becomes approximate away from it.

We note $\left(D_X^{cart,inert}, D_Y^{cart,inert}\right)$ the coordinates of the scene center in the XY plane of the inertial cartesian coordinate system. We note $\left(D_x^{cart,inert}, D_y^{cart,inert}\right)$ these coordinates normalized to a unit vector.

$$raan_{guess} = arctan2\left(D_y^{cart,inert}, D_x^{cart,inert}\right) \tag{23}$$

<u>Anomaly guess</u>

From the definition of the anomaly, we get a good approximation through,

$$\nu_{guess} = D_{lat} - pa \tag{24}$$

10.3.7.3 *Search method*

Since we only consider the ellipsoid in our problem, the function D_{LoS} is smooth and well-behaved in a large neighborhood around the solution. Thus, we recommend using a widely available and robust search method (see for instance the scipy multidimensional root finders, in particular 'broyden1' is the recommended method).
The starting point for the search is set to $(raan_{guess}, \nu_{guess})$. We do not recommend setting bounds, as it can lock the search on extreme points. We recommend wrapping the angles in the following ranges during the search:

$$raan \leftarrow (raan + \pi)\%(2\pi) - \pi \tag{25}$$

$$\nu \leftarrow \nu\%(2\pi) \tag{26}$$

The absolute convergence threshold is set to the parameter D_{tol},

$$\left\|\overrightarrow{D_{LoS}}\right\| < D_{tol} \tag{27}$$

The search is set to continue up to the maximum number of function evaluation f_{ev}. If the number of objective function evaluations exceeds f_{ev}, then the search stops and the module terminates with an *error*.

Upon successful completion of the search, the values $(raan, \nu)$ are obtained. In this fashion, the nominal orbit initial conditions (before propagation) are completely defined. In addition, the module must report in the log the residual D_{LoS} value obtained at the end of the search.

10.4 Orbit propagation

The orbit (position and velocity, not the attitude), is propagated forward and backward from the already computed position at the center acquisition frame. The *propagator* in its *fine* setting is used.

The propagation is made over a set of explicit dates which correspond exactly to the dates of acquisition start for each frame in either the VNIR or SWIR sequence. This set of acquisition dates is given by the input file in section 6.1.3. Note these dates are thus not uniformly spaced.

10.5 LoS guiding law

The LoS guiding law defines the attitude of the satellite during the acquisition sequence. The satellite can have only one attitude, shared by both VNIR and SWIR channel. We choose to establish the LoS guiding law as if for a VNIR acquisition.
The LoS guiding is established in two parts:

1. First the attitude at the center frame is established.
2. Then the attitudes at other frames in the sequence are computed from the TDI pushbroom constraints relative to the initial center frame.

10.5.1 Center frame attitude

The attitude at the center frame is the reference for the whole sequence. Given the nominal orbit position at the center frame, established previously, the attitude must meet the following objectives as per REQ-SAT-150 [AD6]:

1. The center LoS must intersect with the scene center target (this is achieved by the pitch and roll found previously),
2. The detector projected on the ellipsoid must have its columns parallel to the satellite subtrack.

10.5.1.1 Direction target

Our objective for the column direction at the center frame is established by simulating the natural LoS shift from the center frame to the next frame in the sequence, without attitude modification. We proceed as follows, and as illustrated on Figure 10.

1. The satellite attitude is set to the roll and pitch angle used in the previous determination of the initial orbit.
2. The center LoS intersection at this center frame is computed, I_{init} (this should be equal to D in the ideal case).
3. The orbit is propagated forward in time by one VNIR frame time, at the date at index $f_{cen} + 1$ in the *acquisition sequence dates*.
4. The center LoS intersection is computed at this next frame, I_{dir}.

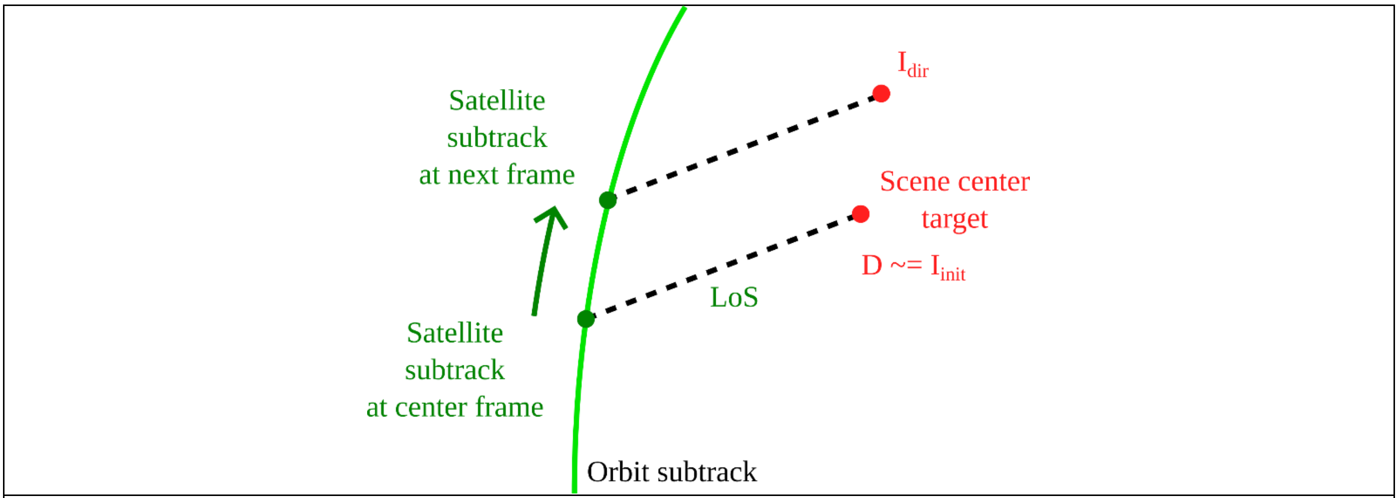

Figure 10 : Illustration of the center frame track direction

We formalize this target direction using the vector formulation presented in appendix 13.1.

$$\overrightarrow{D_{dir}} = \overrightarrow{DI_{dir}} \tag{28}$$

10.5.1.2 Center frame optimization objective

The objective function for the center frame attitude optimization is composed of:

1. The center pointing objective $\overrightarrow{D_{LoS}}$, as defined previously,
2. The angular discrepancy between the target frame direction $\overrightarrow{D_{dir}}$ and actual frame direction $\overrightarrow{I_{init}I_{above}}$.

The actual frame direction is obtained as the vector between I_{init} and I_{above} the LoS intersection of the detector position P_{first}.
The angle between the target direction $\overrightarrow{D_{dir}}$ and the actual direction $\overrightarrow{I_{init}I_{above}}$ is computed as,

$$\alpha_{init} = \arccos\left(\frac{\overrightarrow{D_{dir}} \cdot \overrightarrow{I_{init}I_{above}}}{\left\|\overrightarrow{D_{dir}}\right\| \cdot \left\|\overrightarrow{I_{init}I_{above}}\right\|}\right) \tag{29}$$

Note α_{init} is zero when both vectors are colinear, which is our target.

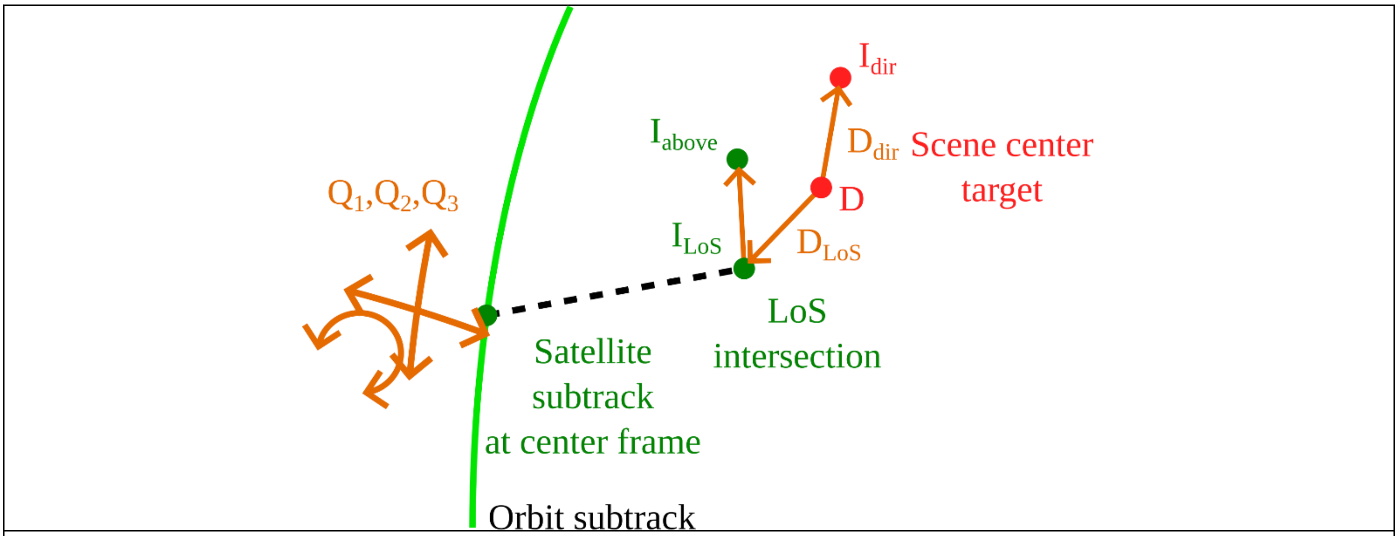

Figure 11 : Illustration of the center frame track direction

Our optimization target is thus to set every component of $(\overrightarrow{D_{LoS}}, \alpha_{init})$ to zero.

10.5.1.3 Attitude optimization

The attitude of the center frame is optimized through a numerical search algorithm of the same kind as that used in section 10.3.7.3. We operate over the raw quaternions with the scalar term $Q_C = 1$, and the complex terms (Q_1, Q_2, Q_3) set as variable. The rationale for using raw quaternions is to simplify and robustify the search. The quaternions will be transformed later to a more reasonable format for output. The solution to our problem is thus,

$$(Q_1, Q_2, Q_3) = \underset{Q_1,Q_2,Q_3}{\operatorname{argmin}}\left(\overrightarrow{D_{LoS}}, \alpha_{init}\right) \tag{30}$$

For the sake of simplicity, we can use $(Q_1, Q_2, Q_3) = (0,0,0)$ as our search starting point, rather than converting the initial roll and pitch to quaternion, which corresponds to the identity rotation with $Q_C = 1$ and thus a pointing aligned with the local vertical in our choice of coordinate system.

The absolute convergence threshold is set to the parameter D_{tol} and α_{tol},

$$\left\|\overrightarrow{D_{LoS}}\right\| < D_{tol} \text{ AND } |\alpha_{init}| < \alpha_{tol} \tag{31}$$

The search is set to continue up to the maximum number of function evaluation f_{ev}. If the number of objective function evaluations exceeds f_{ev}, then the search stops and the module terminates with an *error*.

Upon successful completion of the search, the values (Q_1, Q_2, Q_3) are obtained. In addition, the module must report in the log the residual D_{LoS} and α_{init} values obtained at the end of the search.

10.5.2 Attitude law optimization

The HYP4U satellite requirements REQ-SAT-146 and REQ-SAT-147 [AD6] specify the behaviour of the attitude law over the full acquisition sequence. We translate the requirements as the following objectives, at any given frame of sequence index $f \in [\![0, n_f - 1]\!]$,

1. Align the center LoS intersection of the current frame with the I_{above} intersection of the previous frame. This condition essentially sets the FMC of the manoeuvre together with the frame time.
2. Align the column direction at the current frame $\overrightarrow{I_{cen}I_{above}}$ with the column direction of the previous frame. This condition minimizes the shift of pixels along a TDI sequence, at least in the center of the field.

Let us formalize these objectives, at every frame in the sequence, we consider the following intersection points,

- $I_{cen,f}$ the LoS intersection for the detector center,
- $I_{above,f}$ the LoS intersection for the position above the center in the TDI sequence.

The attitude law is optimized frame by frame, in a forward and backward directions from the center frame. We are looking for a solution (Q_1, Q_2, Q_3) at every frame f.

10.5.2.1 Forward from center frame

Starting from the center frame, we optimize the attitude of frame $f \in [\![f_{cen}+1, n_f-1]\!]$ using the objectives,

1. Align $I_{cen,f}$ with $I_{above,f-1}$ through a metric $\overrightarrow{D_{LoS,f}}$,
2. Align the direction $\overrightarrow{I_{cen}I_{above}}^f$ with $\overrightarrow{I_{cen}I_{above}}^{f-1}$ through the metric α^f.

The optimization target is thus,

$$\forall f \in [\![f_{cen}+1, n_f-1]\!],$$
$$(Q_1, Q_2, Q_3)_f = \underset{Q_1,Q_2,Q_3}{\operatorname{argmin}}\left(\overrightarrow{D_{LoS,f}}, \alpha^f\right) \quad (32)$$

The search must be initialized to $(Q_1, Q_2, Q_3)_{f-1}$.
The absolute convergence threshold is set to the parameter D_{tol} and α_{tol},

$$\left\|\overrightarrow{D_{LoS,f}}\right\| < D_{tol} \text{ AND } \left|\alpha^f\right| < \alpha_{tol} \quad (33)$$

Figure 12 illustrates this optimization.

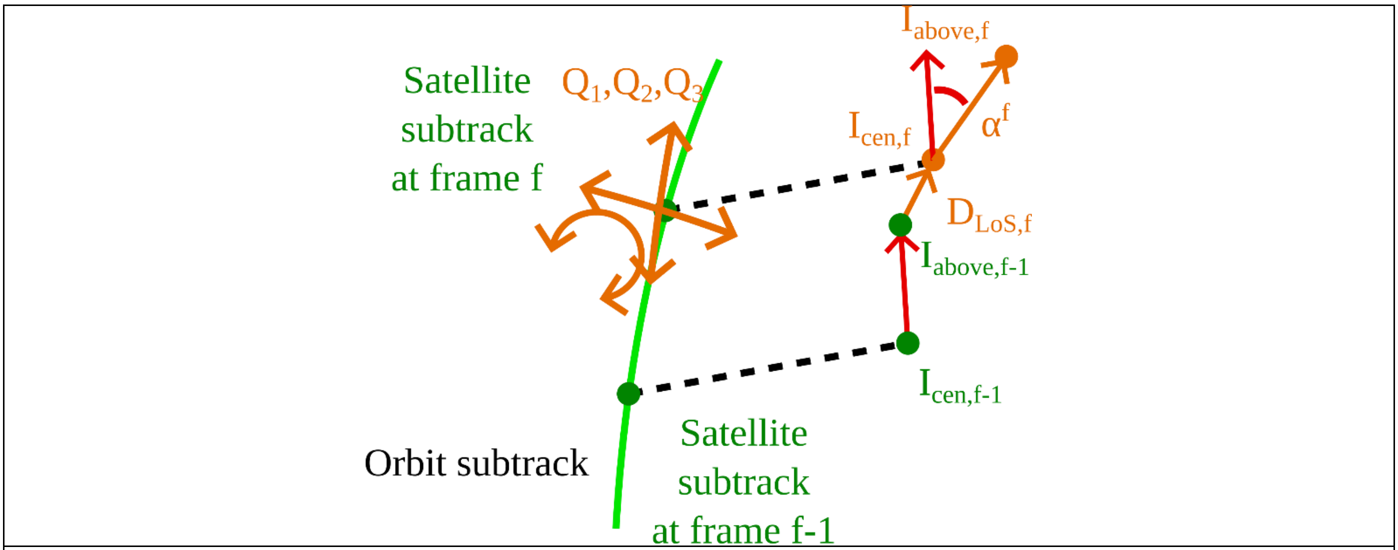

Figure 12 : Illustration of the attitude law optimization in forward direction

10.5.2.2 Backward from center frame

Starting from the center frame, we optimize the attitude of frame $f \in [\![0, f_{cen}-1]\!]$ using the objectives,

1. Align $I_{cen,f}$ with $I_{above,f+1}$ through a metric $\overrightarrow{D_{LoS,f}}$,
2. Align the direction $\overrightarrow{I_{cen}I_{above}}^f$ with $\overrightarrow{I_{cen}I_{above}}^{f+1}$ through the metric α^f.

The optimization target is thus,

$$\forall f \in [\![0, f_{cen} - 1]\!],$$
$$(Q_1, Q_2, Q_3)_f = \underset{Q_1,Q_2,Q_3}{\operatorname{argmin}}\left(\overrightarrow{D_{LoS,f}}, \alpha^f\right) \tag{34}$$

The search must be initialized to $(Q_1, Q_2, Q_3)_{f+1}$.
The absolute convergence threshold is set to the parameter D_{tol} and α_{tol},

$$\left\|\overrightarrow{D_{LoS,f}}\right\| < D_{tol} \text{ AND } \left|\alpha^f\right| < \alpha_{tol} \tag{35}$$

10.5.2.3 Attitude law error metrics

The module shall report in the log the following error metrics relative to the attitude law,

1. Average of $\left\|\overrightarrow{D_{LoS,f}}\right\|$ across the frame sequence,
2. Maximum value of $\left\|\overrightarrow{D_{LoS,f}}\right\|$ across the frame sequence,
3. Average of $\left|\alpha^f\right|$ across the frame sequence,
4. Maximum value of $\left|\alpha^f\right|$ across the frame sequence.

10.6 Real profiles generation

The module generates *real* orbit and attitude sequences given the PSD characteristic of the errors. Note the module shall allow the output of a *real* set of orbit/attitude data with zero error, based on the configuration flags "enable orbit realization error" $orbit^p_{errors}$ and "enable attitude realization error" $attitude^p_{errors}$. In such cases where the errors are disabled, then the real sets are just a copy of the nominal sets.

10.6.1 Generate a random time series with prescribed power spectrum density

We describe a generic method to generate a random time series from a prescribed PSD, in one dimension. This method uses an inverse FFT from spectral to time domain.

10.6.1.1 Method inputs

Let the n_{PSD} prescribed PSD points, as frequency-value pairs, noted,

$$\forall i \in [\![0, n_{PSD} - 1]\!], \{freq_i, PSD_i\} \tag{36}$$

The maximum frequency in this prescribed PSD is noted $freq_{pmax} = \max_i(freq_i)$.

We want to generate a temporal signal with the following characteristics,

- Duration of T_{span},
- Time sampling step of T_{step}.

10.6.1.2 Precomputation

We thus want a temporal signal with n_{time} points.

$$n_{time} = \left\lceil \frac{T_{span}}{T_{step}} + 1 \right\rceil \tag{37}$$

The temporal signal we want is purely real. This means the spectral signal is symmetric, we only need to represent half the spectrum to obtain the full information (plus the zero-frequency component). The number of spectral domain points is $n_{spectral}$.

$$n_{spectral} = \left\lfloor \frac{n_{time}}{2} \right\rfloor + 1 \tag{38}$$

The frequency range taken into account will thus be $[0, freq_{max}]$.

$$freq_{max} = \frac{n_{time} - 1}{2 \cdot T_{span}} \tag{39}$$

The module shall emit a warning if $freq_{max} < freq_{pmax}$, which corresponds to a case where the extremal part of the prescribed PSD will be ignored. This condition can be mitigated by produced a more finely sampled temporal signal.

The frequency sampling step is $freq_{step}$.

$$freq_{step} = \frac{freq_{max}}{n_{spectral}} \tag{40}$$

The prescribed PSD points are linearly interpolated over the range of frequencies $[0, freq_{max}]$ with uniform sampling step $freq_{step}$. The parts of the prescribed PSD which cannot be interpolated are replaced with zeros. We obtain an interpolated PSD,

$$\begin{gathered} \forall \nu \in [\![0, n_{spectral} - 1]\!], \\ freq_{\nu} = \nu \cdot freq_{step} \\ PSD_{\nu}^{inter} = Lin(\{freq_i, PSD_i\}, freq_{\nu}) \end{gathered} \tag{41}$$

With Lin the (loosely defined) linear interpolation operator over the PSD points.

10.6.1.3 *Spectral signal*

We first transform the PSD into a spectral domain signal by taking its square root and assigning it random phases. The phase of the zero-frequency component is set arbitrarily to zero, the rest of the phases are drawn independently from a uniform distribution in the $[0, 2\pi]$ range.

$$\begin{gathered} \phi_0 = 0 \\ \forall \nu \in [\![1, n_{spectral} - 1]\!], \phi_{\nu} = \mathcal{U}_{0,2\cdot\pi} \end{gathered} \tag{42}$$

The discrete spectral signal is obtained as,

$$\begin{gathered} \forall \nu \in [\![0, n_{spectral} - 1]\!], \\ s_{\nu}^{spectral} = \sqrt{PSD_{nu}^{inter} \cdot freq_{step}} \cdot \exp(i \cdot \phi_{\nu}) \end{gathered} \tag{43}$$

ORUS
RCS Grasse 984 607 671 - Siret 984 607 671 00026
NAF 7490 B - TVA # FR 74 984 607 671

10.6.1.4 Time signal

The time signal is obtained by an inverse FFT on the one-sided (Hermitian symmetric) spectral signal[1] (note this is not the most usual interface for a FFT).

$$s^{time} = IRFFT\left(s^{spectral}\right) \cdot n_{time} \tag{44}$$

The result is the discrete, real, set of points $s^{time} = \left\{s_t^{time} \middle| t \in [\![0, n_{time} - 1]\!]\right\}$, the dates during the time span, in correspondence to these value points are,

$$\begin{gathered} \forall t \in [\![0, n_{time} - 1]\!], \\ s_t^{dates} = T_{step} \cdot t \end{gathered} \tag{45}$$

10.6.1.5 Checking the power of signals

The power contained in the PSD and in the generated time signal are the same, this is checked by the module. The power of the PSD is computed by integrating it over the considered frequency range. We use a trapezoidal rule for the discrete integral approximation.

$$\begin{aligned} Pow_{spectral} &= \sum_{v=1}^{n_{spectral}-1} \frac{PSD_{v-1}^{inter} + PSD_v^{inter}}{2} \cdot freq_{step} \\ &\cong \int_{freq=0}^{freq_{max}} PSD \cdot dfreq \end{aligned} \tag{46}$$

The power of the discrete temporal signal is,

$$Pow_{time} = \frac{\sum_{t=0}^{n_{time}-1} \left(s_t^{time}\right)^2}{n_{time}} \tag{47}$$

An error is emitted by the module if the condition $Pow_{spectral} \neq Pow_{time}$ is encountered, taking a small numerical tolerance into account.
As an offline test, the spectrum of the generated time signal can be analyzed and compared to the original PSD.

10.6.1.6 Limitation

Note the presently described method couples the temporal sampling with the PSD sampling. This may skew some of the finer analyses. The method could be extended to independent temporal and spectral samplings by generating multiple random temporal signals over smaller time spans and knitting them back-to-back. The need for this extended method is TBC.

10.6.2 Real orbit generation

The *real* orbit is generated from the *nominal* orbit by using the PSD input in the three LOF axes. A random position error is generated independently over the three axes using the generic method to generate a time series with a prescribed PSD (section 10.6.2). A seed shared over the three axes is used for repeatability.

- $T_{span} = \max_{f \in [\![0, n_f - 1]\!]} T_f^{acq} - \min_{f \in [\![0, n_f - 1]\!]} T_f^{acq}$,
- $T_{step} = \frac{T_{span}}{n_f - 1}$ such that the temporal signal has n_f date points.

[1] https://numpy.org/doc/2.3/reference/generated/numpy.fft.rfft.html is compliant.

This position error is added **in the LOF frame** to the nominal orbit. All required frame conversions must be performed as needed. The output result is reinterpolated over the required output dates.

Note that it is still TBC whether it makes sense at all to get an orbit realization error modelled as a PSD. The real orbit generation may change.

10.6.3 Real attitude generation

The *real* attitude is generated from the *nominal* attitude by using the PSD input, which is a set of rotations in the three LOF axes. A random angular error is generated independently over the three axes using the generic method to generate a time series with a prescribed PSD (section 10.6.1). A seed shared over the three axes is used for repeatability.

- $T_{span} = \max_{f \in [\![0, n_f - 1]\!]} T_f^{acq} - \min_{f \in [\![0, n_f - 1]\!]} T_f^{acq}$,
- $T_{step} = \frac{T_{span}}{n_f - 1}$ such that the temporal signal has n_f date points.

This angular error is added **in the LOF frame** to the nominal orbit. All required frame conversions must be performed as needed. The output result is reinterpolated over the required output dates. The order of rotation is TBD.

11. VISUALIZATIONS

The following visualizations must be outputted by the module.

- Plot of the nominal orbit. This is useful to user, to check that this is the planning they wanted.
  - With the scene target position
  - With the satellite subtrack when pointing at target.
  - With the center roll and pitch indicated.
  - With a long nominal orbit subtrack of a few hours (say 5 hours).
- Plot of the real orbit deviation from the nominal orbit, in all three LOF axes.
- Plot of the nominal attitude, in all three LOF axes.
- Plot of the real attitude deviation from the nominal attitude, in all three LOF axes.
- (TBC) TDI shift of the LOS guiding law for nominal orbit and ellipsoid terrain.

12. DEVELOPMENT PROGRESS

The geometry module is first introduced in the context of the E2EIS v3. We anticipated the inputs which could be provided by the HYP4U platform supplier. The format and content of satellite inputs may still change as development progresses.

13. APPENDIX

We give in appendix a set of default module parameters, and generic methods and conventions.

13.1 Default configuration parameters

We list the default configuration values for the HYP4U-1 instrument. The justification and references are indexed through a dedicated Table 12.

Table 12: Appendix reference documents.

Title	Reference	Revision	Source	Index
Choix du point de fonctionnement de l'instrument	PRJ1970-CRR-0035	A1	ORUS	ARD1
Optical design description ff04	A0003735-DED-0002	A2	SOPHIA	ARD2
CMV12000 datasheet	-	V2.12	CMOSIS/AMS	ARD3

Table 13: Acquisition parameters (VNIR)

Index	Parameter designation	Quantity [unit]	Default value	Justification
ACQ1	Spatial binning factor	-	2	Default acquisition parameter [ARD1]
ACQ2	Latitude of the scene center to acquire	Latitude [degrees]	$43.61335629109762\ deg$	Arbitrarily set to the location of ORUS HQ.
ACQ3	Longitude of the scene center to acquire	Longitude [degrees]	$7.049729051750443\ deg$	Idem
ACQ4	Altitude of the scene center to acquire	Altitude [m]	$125\ m$	idem
ACQ5	Roll angle of the center LOS at the scene center	Angle [degrees]	$20\ deg$	Limit of the roll angle performance range, REQ-SAT-060 [AD6]
ACQ6	Pitch angle of the center LOS at the scene center	Angle [degrees]	$0\ deg$	Target pitch angle for acquisitions, REQ-SAT-065 [AD6]
ACQ7	Repeat ground tracks parameters	See below	$n = 15, q = 28, m = 3$	Arbitrary setting, pending definition [TBD].
ACQ8	Orbit eccentricity	Eccentricity [-]	0	Circular orbit hypothesis
ACQ9	Orbit perigee argument	Angle [rad]	0	Idem, set to zero arbitrarily for circular orbits.

Table 14: Instrument description (VNIR)

Index	Parameter designation	Quantity [unit]	Default value	Justification
IN1	Telescope focal length in the ALT direction	Length [mm]	+ 688.8 mm	[ARD2]
IN2	Detector pixel pitch	Length [µm]	5.5 µm	[ARD3]

Table 15: Simulation parameters

Index	Parameter designation	Quantity [unit]	Default value	Justification
SIM1	Convergence tolerance for the computation of the semi-axis	Length [m]	1 mm	Negligible position error for a satellite
SIM2	Maximum allowable iterations for the initialization of the orbit	#	100	Arbitrary
SIM3	Maximum number of LoS evaluations for the RAAN/anomaly search and attitude optimization	#	100	Arbitrary
SIM4	LoS convergence threshold for the RAAN/anomaly search and for the attitude optimization	Distance [m]	1 cm	Negligible next to GSD
SIM5	Angular tolerance for the frame alignment during the attitude optimization	Angle [rad]	1 µrad	1cm lateral error over 8km for a plane target
SIM6	Flag enabling the orbit realization errors	Flag	True (enabled)	-
SIM7	Flag enable the attitude realization errors	Flag	True (enabled)	-
SIM8	Satellite mass	Mass [kg]	150 kg	Representative mass, pending definition.

13.2 Topographic vector between geodetic points

Given two (neighbor) geodetic points G and H with latitude and longitude coordinates, we want to compute a difference vector commensurate with meters and aligned with the latitude and longitude grid. We use the following topographic projection, which assumes a spherical Earth. This model is only valid for G and H near each other (a few km at most).

In this document, we note the difference vector $\overrightarrow{GH} = (d_{lat}^m, d_{lon}^m)$. It is computed as follows,

$$\begin{aligned} d_{lat} &= H^{lat} - G^{lat} \\ d_{lon} &= H^{lon} - G^{lon} \end{aligned} \tag{48}$$

Then, we reproject this difference over the range $[-\pi; \pi[$,

$$\begin{aligned} d_{lat} &= (d_{lat} + \pi)\%(2\pi) - \pi \\ d_{lon} &= (d_{lon} + \pi)\%(2\pi) - \pi \end{aligned} \tag{49}$$

Then, the distance in meters between the two points is approximated with a spherical Earth model,

$$\begin{aligned} d_{lat}^m &= \tan(d_{lat}) \cdot R_{\oplus} \\ d_{lon}^m &= \cos\left(D^{lat}\right) \cdot \tan(d_{lon}) \cdot R_{\oplus} \end{aligned} \tag{50}$$

13.3 Quaternion conventions

Raw quaternions have too many degrees of freedom when used to only represent attitudes. In order to obtain unique results, we set a number of conventions in place. We note the general quaternion as,

$$\boldsymbol{Q} = (Q_1, Q_2, Q_3, Q_C) \in \mathbb{R}^4 \tag{51}$$

The [AD3] does not require anything for quaternion apart from pure format considerations, so we fall back on [RD12] for recommendations.

13.3.1 Quaternion terms ordering

This convention about terms ordering is purely syntactic. We order the quaternion terms as,

1. Q_1: Component of the first basis vector,
2. Q_2: Component of the second basis vector,
3. Q_3: Component of the third basis vector,
4. Q_C: So-called scalar component.

The first, second and third basis vectors typically refer to the X, Y, and Z vectors of whatever cartesian coordinate system is used to represent the attitudes.

13.3.2 Quaternion normalization

We normalize the quaternion to a *unit quaternion*.

$$||\boldsymbol{Q}|| = \sqrt{Q_1^2 + Q_2^2 + Q_3^2 + Q_C^2} = 1 \tag{52}$$

As per [RD12], such a unit quaternion can be seen as a rotation of angle ϕ around an axis (e_1, e_2, e_3) in the current basis.

$$\begin{aligned} Q_1 &= \sin\left(\frac{\phi}{2}\right) \cdot e_1 \\ Q_2 &= \sin\left(\frac{\phi}{2}\right) \cdot e_2 \\ Q_3 &= \sin\left(\frac{\phi}{2}\right) \cdot e_3 \\ Q_C &= \cos\left(\frac{\phi}{2}\right) \end{aligned} \tag{53}$$

13.3.3 Quaternion disambiguation

Any rotation in 3D space can be represented exactly by two quaternions, $\boldsymbol{Q}$ and $-\boldsymbol{Q} = (-Q_1, -Q_2, -Q_3, -Q_C)$. We did not find recommendations related to this point in either [AD3] or [RD12], so we apply our own rules. To break the ambiguity, we follow the rules, in order,

1. If $Q_C \neq 0$, then choose the quaternion with positive Q_C.
2. If $Q_C = 0$, then
   a. identify the largest remaining component Q_m in absolute value among (Q_1, Q_2, Q_3). If there are multiple components tied for maximum absolute value then choose the first one in syntactic order among the candidates.
   b. Choose the quaternion with Q_m positive.
3. If every component is zero then the quaternion is invalid for our purposes as it does not represent a rotation (not even identity).

-- End of Document --